RJL SOLUTIONS

DELIVERY GUIDE / FIRST GOVERNED CAPABILITY

The 90-Day First Governed Capability

A delivery plan from Frame to an evidence-based investment decision.

FRAME · PROVE · BUILD · RUN & TRANSFER

Richard White
RJL Solutions · rjlsol.com

OCTOBER 2026

01 / THE OBJECTIVE

What the first 90 days should produce.

The first 90 days should establish whether one enterprise data and AI capability deserves continuing investment. Choose a repeated business task. Prove a complete path with real information and controls, then develop the approved slice into a service your team can run. Measure useful work against an agreed baseline, and let that evidence decide the next scope.

This guide is for the people who own that decision: the executive sponsor, the business process owner, the delivery lead and the internal team. It turns Frame, Prove, Build and Run & Transfer into weekly outputs, review gates and ownership commitments.

BY THE REVIEW	EVIDENCE TO HAVE IN HAND
Business decision	Baseline, accepted outcome measure, revised economics, and the next investment decision
Governed capability	Working slice, approved users and scope, tested meaning and authority, request evidence
Who runs it next	Release status, unresolved dependencies, owners, and a funded completion or operation plan
Internal ownership	Engineers involved in delivery, demonstrated operating tasks, remaining transfer conditions

What the window does and does not cover

The published durations are two to three weeks for Frame, four to six for Prove and eight to sixteen for Build per release. This example gives Frame three weeks and Prove five, then starts Build in week nine. Because Build typically takes eight to sixteen weeks per release, it runs past day 90. Run & Transfer begins when approved work reaches real users, and lasts for as many operating cycles as it takes to demonstrate ownership.

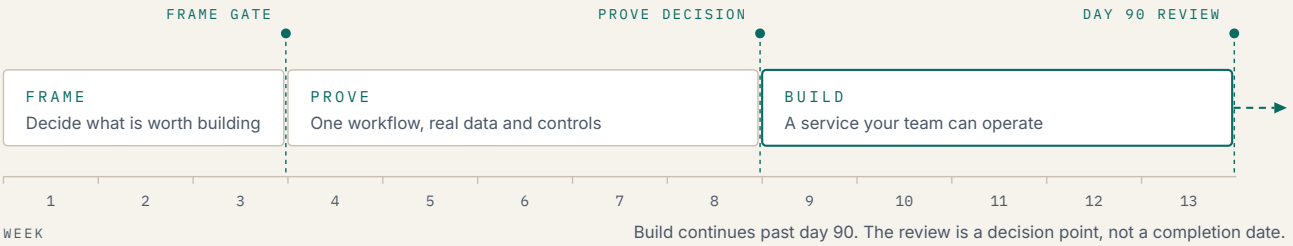
Where approved data products, controls, deployment and support already exist, a narrow capability may reach a controlled live release inside this window. Agree that scope and schedule from the dependency evidence. In an unprepared environment, Week 12 is a readiness review, and the first release follows in Build after day 90.

WHAT DAY 90 STATES	Exactly what is proven, what is running, and what further work is funded. A working demonstration alone does not satisfy release or transfer acceptance.

02 / THE PLAN

The weekly plan and its decision gates.

Use the plan to sequence the decisions and the engineering that shape the first workflow. Business, data, security and internal engineering take part throughout. Identify access and decision holders before the work that depends on them.



RUN & TRANSFER
Begins at the first live release, usually after day 90, and runs until internal ownership is demonstrated.

Thirteen weeks, three gates, and a Build phase that runs past the window. The day-90 review falls in week 13.

PERIOD	OUTPUT AND COMMITMENT AT THE BOUNDARY
Weeks 1 to 3, Frame	Agreed business case, measurement, scope, constraints, target architecture, ownership, and a sequenced plan
Weeks 4 to 8, Prove	End-to-end slice in an approved enterprise environment, evaluation, cost per useful task, revised design, and a go or no-go recommendation
Weeks 9 to 13, Build	Owned products and interfaces, release pipeline, controls, monitoring, support work, and evidence for the actual release state
Day 90, Review	Decision to expand, continue, reshape or stop, with reasons, funding, owners and the next review date

Keep one task complete

Start with a named user group and one repeated task. Include the minimum owned data products, approved definitions and applicable knowledge needed to complete it. Use narrow read tools with a tested identity path. Add relationship modeling only when the question needs it. Leave writes and external communication out of this first scope.

Review dependencies every week. An unavailable source, an unresolved definition or a missing authority changes the plan. Give it an owner and a decision date. If the affected task can’t be proven, reshape the scope.

> **Protect the release criteria when the calendar becomes tight.**

03 / FRAME

Frame sets the commitment, in weeks one to three.

Frame produces a decision about a specific workflow, plus the evidence that would justify Prove. Interview the people who own the process and the people who do it. Work with finance on the business case, and with the control owners on the conditions the capability must operate under.

WEEK AND FOCUS	CONCRETE OUTPUT AND REVIEW CONDITION
Week 1 Observe the work	Current workflow, eligible users, task volumes, baseline sample, exceptions, and the accountable process owner. The owner confirms how work is actually completed, and where value can be captured.
Week 2 Resolve the constraints	Source and knowledge inventory, definition decisions, access path, risk conditions, and dependency owners. Start the model risk, third-party, privacy and security review routes now, with named reviewers and lead times. The team can identify the deciding assumptions and obtain authority to test them.
Week 3 Agree on the target	Bounded target architecture, measures and acceptance criteria, evaluation plan, operating owner, and sequenced delivery plan. Name the change leader and agree on the one-page alignment of what changes for whom. The sponsor and decision holders agree on the scope and the next commitment.

The Frame packet holds the business case with its agreed measurement, the constraint map, the target architecture and the sequenced plan. Attach the owner and decision register, the exclusions, the evaluation criteria and the transfer conditions, so the next phase works from the same commitments.

Build a baseline that survives review

Measure task completion the way the business experiences it, including verification, corrections, waiting and exceptions. Keep the sample definition, the time period, the eligible population and the known limitations. Confirm how later comparisons will handle changes in case mix or workload.

Time released doesn’t automatically become cash savings. Finance should separate capacity, avoided expense, service improvement and benefit that needs a process or staffing change. Record how the owner expects to realize value, and what it will cost to maintain.

GATE AT THE END OF FRAME	Proceed when the business owner accepts the baseline, can change the workflow, and can make the deciding dependencies available for testing. Reshape when a narrower task has a credible path. Stop when value, authority, access or operating capacity cannot support the investment. Do not start Prove with an unowned access request or an undefined standard for success.

Decide where AI should not act before scoping the build.

Frame also decides how much the capability may do. Two questions set the bound for each task: what would an error cost, and how much human judgment does the task need? Each quadrant carries a control.

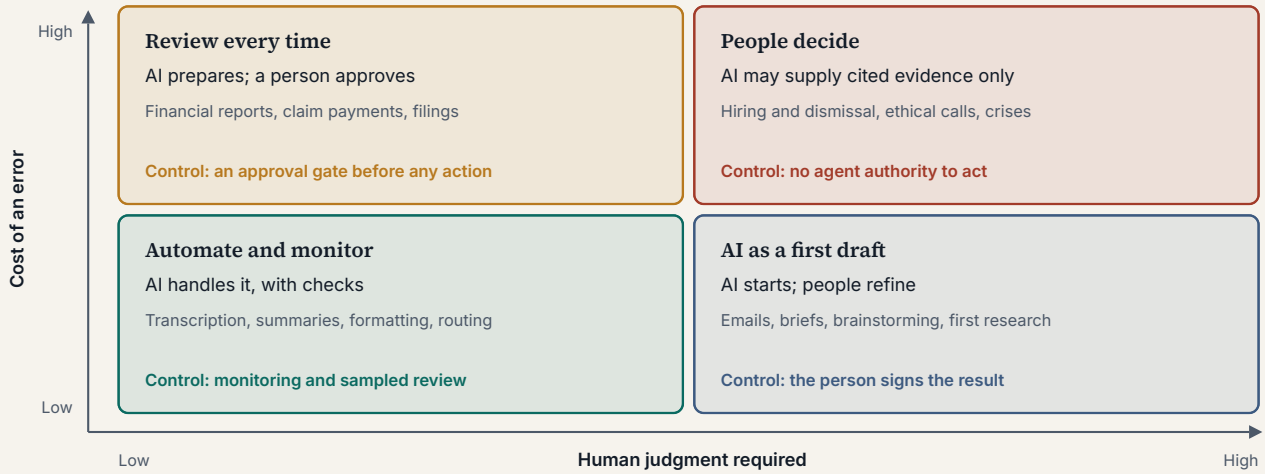


Exhibit: The cost of an error and the judgment required decide how much AI may do.

In week two, place each candidate task in a quadrant with the business owner and the control owner. Tasks where people decide stay outside the agent's authority. Tasks that need review every time carry their review capacity and cost into the business case, so the baseline and the target compare the same work.

04 / PROVE

Prove connects the workflow, in weeks four to eight.

Prove uses real enterprise information in an approved enterprise environment. It tests business meaning, usefulness, permissions and economics along the complete request path. Internal engineers work with the delivery team on the implementation and the evidence from the first week of this phase.

WEEK AND FOCUS	CONCRETE OUTPUT AND REVIEW CONDITION
Week 4 **Establish the dependencies**	Scoped source access, classification, owned product contracts, approved definitions, and applicable knowledge. Confirm effective dates, provenance, freshness needs, and the identity used to reach each dependency.
Week 5 **Connect the request path**	Working read interfaces, retrieval where needed, typed tool contracts, and a correlated trace. Demonstrate that entitlement holds at execution, and through the relevant caches or derived artifacts.
Week 6 **Complete representative tasks**	End-to-end workflow with source context, usable answers, incomplete states, and business review. Real users evaluate their own cases against accepted examples; record defects by consequence and affected scope.

Take an illustrative claims-information workflow (see *The Four-Part System*, section 09). Connect the permitted claim population to approved entities and relationships, the designated status source and the applicable procedure. The answer separates facts from interpretation and flags missing or conflicting evidence. An authorized reviewer can inspect the request's identity, versions, sources, policy outcome and result.

Test the constraints as the path is built

Test disallowed records, revoked users, stale information, missing knowledge, ambiguous identifiers and conflicting relationships. Plant hostile instructions inside retrieved material. Those passages supply evidence for the task. They must never grant tools or authority.

A successful request proves only its tested conditions. Keep the proposed population, the important exceptions and the limits visible in the evaluation. A narrow tool needs authorization in the execution path, even when someone instructs the agent to stay within scope.

Keep the design responsive to evidence

Keep a defect and decision register with an owner, impact, closure criterion and proposed resolution for each entry. When results expose a bad assumption, revise mappings, contracts, knowledge selection or the workflow. Keep the original acceptance criteria, and explain any agreed change.

05 / THE PROVE DECISION

Weeks seven and eight turn a slice into a recommendation.

Keep the evaluation cases separate from the examples used to tune the implementation. Freeze the candidate artifacts for the review. Keep the identities, scope, versions and source context that qualify the results.

Week 7 tests usefulness and failure behavior

Evaluate representative work, material exceptions, authority boundaries and dependency failures. Review results with the process owner and the relevant control owners. Capture user verification and correction effort, not just system latency. Measure full cost per useful task under stated volume and concurrency assumptions.

Exercise unavailable identity or policy, exhausted limits, failed tools and stale sources. Check that the task fails safely, or returns the agreed incomplete state. A fallback has to keep permissions and show its limitations. A critical failure in disclosure, unauthorized action, isolation or required evidence blocks the affected release scope.

Week 8 revises the case and makes the decision

Demonstrate the slice with selected requests and their evidence, alongside the aggregate results and the failures. Update the architecture, dependency plan, economics and Build estimate. The recommendation should say which users, information and operations its evidence supports, and which assumptions still need work.

REVIEW PACKET	WHAT THE DECISION HOLDER SHOULD BE ABLE TO ASSESS
Working slice and request trace	Complete task, sources, meaning, identity, policy, versions, and the supported result
Evaluation report	Coverage, useful completion, important errors, boundary tests, and unresolved defects
Economic readout	Baseline comparison, verification effort, full operating cost, and benefit assumptions
Revised delivery plan	Component changes, dependencies, release scope, support, transfer, and completion estimate

PROCEED TO BUILD WHEN	Value remains credible, the deciding constraints are proven or have an accepted closure path, and the team can fund the intended operating destination. Prove approval authorizes the next delivery commitment. Production acceptance remains a separate decision.

06 / BUILD

Build the accepted slice, in weeks nine to thirteen.

Build turns the accepted slice into a maintained service. The published Build range is eight to sixteen weeks per release; in this example, weeks nine to thirteen cover its first five weeks. Schedule the rest against the actual dependencies.

WEEK AND FOCUS	CONCRETE OUTPUT AND REVIEW CONDITION
Week 9 The release contract	Confirm users, operations, product owners, dependencies, service objectives and backlog. Link the approved scope and evaluations to versioned release artifacts.
Week 10 Make delivery repeatable	Test and deployment pipeline, environment configuration, controlled identities, dependency validation, and change review. Reproduce the reviewed release in the intended environment.
Week 11 Establish service behavior	Monitoring, task and cost limits, incident routing, revocation, restriction and recovery procedures. Internal owners exercise the procedures with their own access.
Week 12 Review release readiness	Inspect regression results, enforcement, dependency commitments, support, training and revised roles, and any required model risk or validation sign-off. Release only the scope its evidence supports; otherwise continue in approved testing.
Week 13 Review the actual state	If live, examine controlled operating cycles, exceptions, corrections and cost. If testing, inspect the remaining Build work. Prepare the day-90 decision and the funded next plan.

Release by evidence, and operate by scope

A release is a named combination of versions: model, prompts, retrieval, definitions, tools, policy and workflow. The release packet links that combination to passing evaluation, approvals, required contracts, operating limits, monitoring, support and recovery evidence. Confirm what users should do when information is incomplete, access is denied, a source is stale or the service is down.

The scope should say what the capability can read, and what it can’t do. Extending it to writes, a new population, another source or external communication needs assessment and approval of the changed behavior.

Run begins when approved work reaches real users. Observe with the internal team through the cycles that matter to the service. A production date doesn’t mean transfer is complete. And recovering the software doesn’t reverse a business action that already happened.

07 / OWNERSHIP

Make ownership part of the delivery plan.

Assign decision, delivery and service owners during Frame. A small team can combine roles while keeping authority explicit. Receiving engineers and service owners take part throughout delivery.

OWNER OR ROLE	RESPONSIBILITY FOR THE FIRST CAPABILITY
Executive sponsor and finance	Investment boundary, benefit mechanism, continuing funding, and the next commitment
Business process owner	Workflow changes, eligible users, useful outcome, adoption, exceptions, and acceptance
Domain and product owners	Definitions, authoritative sources, quality, freshness, applicable knowledge, change notice
Delivery and platform owners	Environments, integration, release pipeline, service commitments, and technical recovery
Security and relevant controls	Permitted use, enforcement conditions, review authority, and incident coordination
Internal service owner	Support, operating cadence, change review, maintenance capacity, and accepted transfer

The capability owner coordinates the work. The relevant decision holders keep authority over access, definitions and risk. Record an escalation route for disputed meaning, blocked access and unmet dependencies.

Establish a working cadence

Each week, review evidence, material defects, dependencies and the decisions due, with the relevant business and control owners. Give every gap that matters an owner and a closure criterion. Decision holders assess the same phase packet. During operation, review incidents and dependencies by criticality, and review adoption, outcomes and recurring cost with the business owner.

Demonstrate transfer before accepting it

Have the receiving team demonstrate user admission and revocation, request investigation, change review, restriction and recovery. Verify access, authority, skills and funded capacity. Keep runbooks, support assignments and skills and staffing plans current. Record the remaining transfer conditions and their owners. A repository or a training record alone isn’t enough.

Who is in the room.

This is a representative structure. It isn't a fixed staffing model. The Frame proposal sets the team and the time needed from business, technology and control owners, and both change with the workflow, the risk involved and what's already in place.

RJL ROLE	MAIN WORK	WHEN
Richard White, principal	Accountable for the outcome; leads Frame, the sponsor relationship and the gate reviews	Throughout
Engagement or product manager	Plan, backlog, gates and client coordination	Full time
Solution and data architect	Target design, semantic layer, data products and identity design	Mostly Frame and Prove
Analyst	Baseline, workflow mapping, measures, and test cases with the experts	Full time
Data engineers, one or two	Data products, contracts and pipelines	Prove and Build
AI engineer	Retrieval, agent and tools, and the evaluation setup	Prove and Build
Security and identity engineer	Access path and revocation tests	Part time
Risk and controls lead	Opens the model risk, compliance and privacy routes in Week 2, and prepares the evidence	Part time
Change and adoption lead	Alignment, listening rounds and training	Part time, from Week 3

Richard is fully involved in the deliverables. You receive the name and role of everyone on the team in the Frame proposal before you sign.

07 / OWNERSHIP, CONTINUED

What the work asks of your team.

Your team does more than review. The owner, subject-matter experts and engineers shape the result, and the operating team carries it after transfer. Indicative time, adjusted in the Frame proposal:

YOUR ROLE	INDICATIVE TIME
Executive sponsor	About an hour a week, plus the three gate reviews
Business process owner	Four to six hours a week
Two or three subject experts	Two to four hours a week each, more in weeks six to eight
Data owner or steward	Two to three hours a week, for definitions
Your engineers	Embedded from Prove, ideally half time or more, so the hand-over is real
Security or identity contact	About two hours a week, plus access setup
Risk, compliance or model risk reviewer	Review route set up early, then the gates
Change leader	Two to four hours a week from Week 3

Prerequisites. A named sponsor and business owner, an environment with approved access to an AI model, data access requests started in Week 1, and a signed confidentiality and services agreement.

Measure useful work, and enforce release conditions.

Frame sets the measures, populations, targets and observation windows. Prove tests them under declared conditions. Run measures the approved service as people use it. Compare like tasks and case mix, show the sample and its limitations, and keep business outcomes separate from technical service performance.

MEASURE	DEFINITION AND INTERPRETATION TO AGREE
Useful completion	Tasks accepted by the business under the stated criteria, divided by eligible attempted tasks; show incomplete and escalated work separately
Time per useful task	Time from task start to accepted completion, including verification, correction and relevant waiting; compare equivalent tasks
Material error and correction	Errors by consequence and affected population, plus the effort to correct them; distinguish minor defects from release blockers
Adoption and outcome	Actual eligible users and repeat use, alongside the workflow result the owner intends to change
Service performance	Availability, latency, freshness and recovery commitments, with windows, dependencies and failure response
Cost per useful task	Recurring run cost for the window (inference, platform, support, review and failed attempts) divided by accepted useful completions; report amortized delivery cost as a separate line

Separate one-time delivery cost from recurring service cost, while showing both in the business case. Declare allocation assumptions for shared infrastructure.

> **When no task meets the useful-completion criteria, report the cost and zero completions.**

Do not present a favorable cost per task from a narrower denominator. Judge each named release on business, control and operational evidence at the same time.

Business results

Useful completions, review load, exceptions, net capacity, full cost

Control results

Access denials, stale permissions, isolation, prohibited actions

Operational results

Load, latency, outages, recovery, rollback and support

Judge the same named release, a fixed combination of versions, on all three. A favorable average cannot override a failed critical control.

Exhibit: Release evidence covers the whole workflow.

Make the acceptance conditions explicit.

Agree the business quality floor, the maximum verification burden, the cost bounds, the service commitments and the control evidence the scope needs. Set them from the workflow's consequences and its baseline. Don't borrow a universal percentage from a generic checklist.

One illustrative access condition: a revoked caller can't reach restricted information through tools, retrieval, sources or the relevant caches. One evidence condition: an authorized reviewer can reconstruct the request's decisions, source context, versions and outcome. Test failures as well as successful requests.

THE RULE THAT DOES NOT BEND	No average accuracy score offsets a critical access, isolation, unauthorized-action, integrity or required-evidence failure. Restrict the affected scope, resolve the cause, and repeat the relevant evaluation before release. Record any permitted exception with its authorized decision holder, conditions, expiry and enforcement.

Make the day-90 decision concrete

The review should establish the capability's actual state, and authorize a specific next commitment. Bring the business case and baseline, the Prove results, the release and operational evidence, the unresolved dependencies and the internal ownership demonstration. In the decision record, keep a tested slice, an accepted live service and completed transfer distinct.

OUTCOME	COMMITMENT AND EVIDENCE TO RECORD
Expand	The next workflow or scope, reused foundations, new risks, operating capacity, and evaluation required
Continue	The remaining Build or operating work, gap owners, closure criteria, funding, and next review date
Reshape	The changed workflow or assumptions, revised value and scope, and the evidence required to proceed
Stop	Why further investment is unsupported, and ownership of access removal, retained artifacts and orderly closure

09 / BEFORE THE FIRST CALL

A brief for the working session.

One page is enough. Bring it to the first conversation, and the session can be about the decision instead of orientation.

01 **The work.** The repeated task, the user population, the process owner, the current baseline, and the expected benefit mechanism.

02 **The dependencies.** The authoritative sources, the applicable knowledge, the read or action boundaries, and the deciding constraints.

03 **The authority.** The people who approve meaning, access, release and continuing investment.

04 **The starting position.** Which existing foundations can be reused, and which gaps need Build.

05 **The destination.** The internal operating owner, the demonstration needed for transfer, and the scope of the day-90 commitment.

06 **One representative request,** and the exception most likely to decide whether the capability is useful.

RJL works through Frame, Prove, Build and Run & Transfer, with ownership designed into delivery. The first conversation should establish whether the workflow has a credible path, and what Frame has to resolve. Each phase is separately scoped and priced in a written proposal before you commit, with its dependencies, acceptance conditions and planned ending.

Request a free sixty-minute working session at **rjlsol.com/contact** or write to **rwhite@rjlsol.com**.

Basis RJL approach and phase durations: rjlsol.com, How we work. The weekly outputs and gates in this guide are a proposed planning pattern to tailor during Frame, not fixed contractual milestones or client results.

Companions The *Connected Enterprise Data & AI Blueprint* for the wider foundation, *The Four-Part System* for architecture responsibilities, *Agent Onboarding* for agent and tool contracts, *Adoption by Design* for the people work in each phase, and the *10-Question Diagnostic* to surface consequential gaps before committing the next scope.

ABOUT

Richard White

I was global firm chief data officer at McKinsey & Company, owning data, AI and platforms end to end, and before that chief data officer at New York Life, where I built the chief data office and reported data risk to risk committees at multiple levels. Earlier I led data at the Federal Reserve Bank of New York and enterprise data governance at Citigroup. I founded RJL Solutions in 2024. My book, *The Day After the Demo*, is free at rjlsol.com/book.

RJL Solutions connects strategy, architecture, engineering, governance and operation through Frame, Prove, Build, and Run & Transfer. Each phase is separately scoped and priced in a written proposal before you commit. The first sixty-minute working session is free.

GUIDE **The Connected Enterprise Data & AI Blueprint**
The wider foundation this plan delivers against, in 24 pages.

ARCHITECTURE **The Four-Part System**
The responsibilities and contracts behind the capability, in 16 pages.

WORKSHEET **The 10-Question AI Production-Readiness Diagnostic**
Four pages to surface the consequential gaps before committing the next scope.

RJL SOLUTIONS

WEB rjlsol.com

EMAIL rwhite@rjlsol.com

This guide describes no client engagement and no client result. The claims workflow is illustrative. The weekly outputs are a planning pattern to tailor during Frame; they aren't fixed contractual milestones. Before production adoption, any implementation has to validate platform maturity, regional availability, licensing, regulatory obligations, performance and operational ownership.